IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	09/848,713
Applicant	:	Doug Grumann
Filed	:	May 3, 2001
Title	:	METHOD AND APPARATUS TO EXTRACT THE HEALTH OF A SERVICE FROM A HOST MACHINE
TC/A.U.	:	2142
Examiner	:	Benjamin A. Ailes
Docket No.	:	10002681-1
Customer No.	:	022879

Mail Stop Appeal Brief - Patents
Commissioner of Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

CORRECTIVE APPEAL BRIEF UNDER 37 C.F.R. §41.37

Dear Sir:

This is in response to the Notice of Non-Compliant Appeal Brief dated January 11, 2008. The enclosed Corrective Appeal Brief Under 37 C.F.R. §41.37 is submitted to correct deficiencies found on sections III and V of the instant brief.

1

TABLE OF CONTENTS

Page

I. REAL PARTY IN INTEREST

Hewlett Packard Company is the real party in interest.

II.   RELATED APPEALS AND INTERFERENCES

There are no other related appeals or interferences.

4

III.   STATUS OF CLAIMS

Claims 1, 2, 4 - 8, 11, 12, 14 - 22 and 24 are pending. Claims 1, 2, 4 - 8, 11, 12, 14 - 22 and 24 are rejected. Claims 3, 9, 10, 13, and 23 are cancelled. Applicant appeals the rejection of claims 1, 2, 4 - 8, 11, 12, 14 - 22 and 24.

IV. STATUS OF AMENDMENTS

There are no amendments filed after final.

6

V. SUMMARY OF CLAIMED SUBJECT MATTER

Applicants have discovered a new, improved method, and a corresponding mechanism, for determining the service health of a computer system. The method includes gathering performance information and translating the gathered performance information, into generic health metrics. The result is an abstracted set of consistent service health metrics that can be provided to the performance monitoring tools such that the performance monitoring tools may use the health metrics without needing to know how the health metrics were derived. This method decouples the performance tool implementation from the metric derivation and removes dependencies between the services, their implementation, and the management tool set. For example, a tool may use the generated service level violation metric to generate an alert when violations raise above a threshold. The performance monitoring tools do not need to know anything about the service, its instrumentation, or how the service level metric is calculated. The tool simply compares the resultant metric against its threshold. The performance monitoring tool uses a programmatic interface library call or script interface to access health metrics for all current services. If the underlying application changes, the current version of the performance monitoring tool is unaffected because of this consistent interface. As a result, the system administrator does not necessarily need to install a new version of the performance monitoring tool. Thus, the apparatus and method are extensible without propagating a dependency up into the higher levels of the management software.

Referring to Figure 3 and its accompanying description starting at page 9, line 20, a system 100 includes a health generator 10 that provides an output of service health data 14 to performance monitoring tools 13. The service health metrics may be directly measured or derived from applications, processes and thread instrumentation, for example. The method is independent of specific provider applications and management tool sets, thereby allowing for shorter time-to-market for service management solutions. The output 14 of the method may be either in the form of a programmatic or scriptable interface 108 to be used by the monitoring tools 13, which are capable of reporting status of many disparate computer services. The tools 13 may reside on different systems and architectures and may be supplied by different vendors. To accommodate different performance monitoring tools, the interfaces are generic and flexible.

Considering the invention recited in claim 1, and referring to Figures 3 and 6, a method 200 for dynamically determining the health of a service resident on a host machine includes collecting 203 service performance information 12 from the resident service,

7

wherein the collected service information relates to a plurality of performance metrics; and translating 207 the collected service performance information into a generic output 14 relating to current operational performance of the service, wherein the generic output 14 is one of a scriptable interface and an application programming interface 108, and useable by different performance monitoring tools 13. See page 9, line 20 to page 10, line 24.

Considering the invention recited in claim 11, and referring to Figures 3 and 4, a health generator 10, which is used to determine a health of a service resident on a host machine, includes a data collection engine 121 that collects service health information 12 and a translation data analysis engine 123 that translates the collected service health information into a generic health output 14 in the form of an application programming interface or a scriptable interface 108 and useable by performance monitoring tools 13. See page 9, line 20 to page 11, line 16.

Considering the invention recited in claim 18, and referring to Figures 3 and 6, a method 200 for monitoring health data of a service operating on a host machine includes collecting 203 service performance information from the service; collecting 203 external performance information from components of the host machine; translating 207 the collected service and external performance information according to a health generation algorithm to generate a generic service health output 14; and providing 209 the generic service health output 14 relating to current operational performance of the service as an output file accessible by performance monitoring tools, wherein the generic service health output is one of a scriptable interface and an application programming interface 108, and usable by the different performance monitoring tools 13. See page 9, line 20 to page 10, line 24 and page 20 line 20 to page 15, line 21.

Considering the invention recited in claim 21, and referring to Figures 3 and 4, an apparatus 10 determines a health of a service, wherein the service operates on a host computer 100, the apparatus 10 including a collection module 121 that receives performance information related to the service; a translation health generator module 125 that applies a rule set 127 to the received performance information and derives generic health metrics 14 therefrom; and an output module 125 that outputs the generic health metrics relating to current operational performance of the service, wherein the generic health metrics are in a format usable by different performance monitoring tools 13. See page 9, line 20 to page 11, line 16 and page 12, line 19 to page 14, line 19.

8

VI.   GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1, 2, 4 - 6, 8 - 12, 14, 15, 17 - 22 and 24 are rejected under 35 U.S.C. § 103(a) over U.S. Patent 6,876,988 to Helsper (hereafter Helsper) in view of U.S. Patent 6,816,898 to Scarpelli (hereafter Scarpelli) and further in view of U.S. Patent 7,020,697 to Goodman et al. (hereafter Goodman).

Claim 7 is rejected under 35 U.S.C. § 103(a) over Helsper, Scarpelli, Goodman and further in view of U.S. Patent 5,949,976 to Chappelle (hereafter Chappelle).

Claim 16 is rejected under 35 U.S.C. § 103(a) over Helsper, Scarpelli, Goodman and further in view of U.S. Patent 6,647,413 to Walrand et al. (hereafter Walrand).

More specifically, and considering independent claims 1, 11, 18, and 21, the Examiner asserts, *inter alia*, that Helsper teaches, at column 2, lines 42 - 60, "translating the collected service information into a generic output relating to current operational performance of the service" but admits that Helsper does not "explicitly recite 'wherein the generic output is accessible by one of a scriptable interface and an application programming interface.'" The Office Action then asserts that Scarpelli teaches the use of "custom script programs but does not explicitly recite wherein the generic output is actually one of a scriptable interface or an application programming interface." The Examiner then states: "Taking its broadest reasonable interpretation in the art, the output data is interpreted when being translated as being any data type that can be processed by computer means." Next, the Examiner states "Goodman teaches wherein data can be translated from specific into a generic API and therefore readable by a plurality of different applications." The Examiner concludes "in view of Goodman, it would have been obvious ... to use a script interface or an application programming interface when the reading in of performance information is necessitated."

VII.   ARGUMENT

CLAIM 1

Considering claim 1, the Examiner asserts that Helsper teaches all that is recited, including that Helsper teaches, at column 2, lines 42 - 60, "translating the collected service information into a generic output relating to current operational performance of the service" but admits that Helsper does not "explicitly recite 'wherein the generic output is accessible by one of a scriptable interface and an application programming interface.'" The Office Action then asserts that Scarpelli teaches the use of "custom script programs but does not explicitly recite wherein the generic output is actually one of a scriptable interface or an application programming interface." The Examiner then states: "Taking its broadest reasonable interpretation in the art, the output data is interpreted when being translated as being any data type that can be processed by computer means." What point the Examiner is making with this statement is unclear. Then, the Examiner states "Goodman teaches wherein data can be translated from specific into a generic API and therefore readable by a plurality of different applications." The Examiner concludes "in view of Goodman, it would have been obvious ... to use a script interface or an application programming interface when the reading in of performance information is necessitated."

Over several Office Action responses, Applicants clearly have demonstrated that Helsper DOES NOT DISCLOSE OR SUGGEST "translating the collected service performance information into a generic output relating to current operational performance of the service." Helsper is directed to a forecasting system that produces near-term predictions of future network performance of e-business systems and system components. However, the output of Helsper's system is not a generic output. Instead, Helsper uses monitoring and forecasting kernels to tailor a concurrent-learning information processor (CIP) to various physical applications. The kernels "may correspond to a spatial configuration of inputs and outputs, a temporal configuration of inputs and outputs, or a combined spatial and temporal configuration of inputs and outputs. As specific examples, a spatial configuration my be tailored to an image processing application, a temporal configuration may be tailored to a commodity price fore-casting application ... ." See column 7, lines 53 - 62. Furthermore, Helsper's system includes a self-learning function wherein output features are, over time, transformed "into computed output values in accordance with output feature specifications supplied by the manager." See column 8, lines 39 - 43. Thus Helsper actually teaches away from a generic output - one of Helsper's key features is an output that changes with each monitoring/forecasting cycle.

10

Scarpelli is directed to a system that collects performance management information. Scarpelli, however, does not cure the defects in Helsper in that Scarpelli does not disclose or suggest a generic output relating to current operational performance of a service, as recited in claim 1. Rather, Scarpelli uses APIs to integrate custom scripts, not generic scripts. See column 4, lines 57 - 60.

Goodman is directed to a computer architecture for a netcentric computing, and more specifically software routines that allow greater flexibility and performance from a distributed computer system. One aspect of Goodman's architecture is computer-telephone integration messaging service 204, which integrates telephone and computer messaging. One function of the service 204 is to provide message mapping. In this regard, the service 204 provides for translating device-specific communications to "generic API and/or message sets." See column 88, lines 10 - 30. Goldman provides no further disclosure to define or suggest the intended purpose of the messages sets. That is, Goodman does not disclose or suggest "useable by different performance monitoring tools."

In contrast to Helsper, Scarpelli, and Goodman, claim 1 recites a "generic output." The feature of the "generic output" is defined in the specification, for example, at page 5, lines 3 - 20:

> However the performance information is gathered, the apparatus and method translate the gathered performance information, or metrics, into health metrics. The result is an abstracted set of consistent service health metrics that can be provided to the performance monitoring tools such that the tools may use these metrics without needing to know how the health metrics were derived. This decouples the performance tool implementation from the metric derivation and removes dependencies between the services, their implementation, and the management tool set. For example, a tool may use the generated service level violation metric to generate an alert when violations raise above a threshold. The performance monitoring tools do not need to know anything about the service, its instrumentation, or how the service level metric is calculated. The tool simply compares the resultant metric against its threshold. The performance monitoring tool uses a programmatic interface library call or script interface to access health metrics for all current services. If the underlying application changes, the current version of the performance monitoring tool is unaffected because of this consistent interface. As a result, the system administrator does not necessarily need to install a new version of the performance monitoring tool. Thus, the apparatus and method are extensible without propagating a dependency up into the higher levels of the management software.

Also note page 7, lines 12 - 23:

> To solve these problems, a method and an apparatus are used to derive consistent service health measures by combining various instrumentation from both internal sources and external sources that relate to the service under observation. The service

11

health metrics may be directly measured or derived from the applications, processes and thread instrumentation, for example. The method is independent of specific provider applications and management tool sets, thereby allowing for shorter time-to-market for service management solutions.

The output of the method may be either in the form of a programmatic or scriptable interface to be used by high-level monitoring tools that are capable of reporting status of many disparate computer services. The tools may reside on different systems and architectures and may be supplied by different vendors. To accommodate different performance monitoring tools, the interfaces are <u>generic</u> and flexible.

Finally, note page 13, lines 15 - 17:

The rules set 127 provides algorithms and rules to translate the metrics supplied by the data collection engine 121 into a <u>generic</u> format that is usable by the performance monitoring tools.

As the above-quoted passages from the specification demonstrate, the term "generic output" is an interface that allows the health metrics to be used without any need to know how the health metrics were derived, and that removes any dependencies between the services, their implementation, and the management tool set. As thus defined, the claimed generic output has a specific meaning that is not addressed in Helsper, Scarpelli and Goodman.

On page 4 of the Office Action, the Examiner states; "taking its broadest reasonable interpretation in the art, the output data is interpreted when being translated as being any data type that can be processed by computer means." What does this mean? Is the Examiner restating an earlier point that a generic output can be any data? (See December 12, 2005 Office Action.) If so, then the modifier "generic" can have no meaning. Instead, the modifier "generic," given its ordinary dictionary definition, should lead to an interpretation of "generic output" as one that has the characteristics of the entire group of outputs. The "any data type" interpretation has a very different meaning, namely any output, <u>regardless</u> of its characteristics. Furthermore, the doctrine of broadest reasonable construction of a claim term still requires that such interpretation "be consistent with the one that those skilled in the art would reach [and must be] consistent with the specification." *See In re Cortright*, 165 F.3d 1353, 49 USPQ2d 1464 (Fed. Cir. 1999). Referring back to the passages from the specification quoted above, clearly, the Examiner's interpretation is neither consistent with the specification nor consistent with that of one of ordinary skill in the art. Thus, the Examiner's interpretation of "generic output" is improper.

Because Helsper, Scarpelli and Goodman, individually and in combination, do not disclose all the features in amended claim 1, claim 1 is patentable.

CLAIM 11

Independent apparatus claim 11 recites a data analysis engine that translates the collected service health information ... and provides one or more generic health metrics, wherein the generic output comprises a plurality of service health metrics, and wherein the translation data analysis engine combines one or more of the plurality of performance metrics to provide one or more of the plurality of service health metrics, and wherein the plurality of service health metrics comprises availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction. As noted above, Helsper does not disclose or suggest all of these features. Accordingly, claim 11 also is patentable.

CLAIM 18

Independent method claim 18 recites the step of generating a generic service health output, wherein the generic output comprises a plurality of service health metrics, and wherein the translation data analysis engine combines one or more of the plurality of performance metrics to provide one or more of the plurality of service health metrics, and wherein the plurality of service health metrics comprises availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction. For the same reasons that claims 1 and 11 are patentable, claim 18 is also patentable.

CLAIM 21

Independent apparatus claim 21 recites service health metrics comprising availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction. As noted above with respect to claim 1, Helsper, Scarpelli, and Goodman do not disclose or suggest these features. Accordingly, claim 21 is also patentable.

DEPENDENT CLAIMS

Claims 2, 4 - 7 and 8 depend from patentable claim 1; claims 12 and 14 - 17 depend from patentable claim 11; claims 19 and 20 depend from patentable claim 18, and claims 22 and 24 depend from patentable claim 21. For these reasons and the additional features they recite, claims 2, 4 - 8, 12, 14 - 17, 19, 20, 22, and 24 also are patentable.

13

The appeal brief fee in the amount of **$510.00** has already been submitted on December 17, 2007. However, should there be any additional fees required for this appeal brief, please charge any fees required or credit any over payment to **Deposit Account 08-2025** pursuant to 37 C.F.R. § 1.25.

Respectfully submitted,

Date: **February 11, 2008**

John K. Harrop, Reg. No. 41,817
ANDREWS KURTH LLP
1350 I Street, NW
Suite 1100
Washington, D.C. 20005
Telephone: (202) 662-3050
Fax: (202) 662-2739

14

VIII. CLAIMS APPENDIX

1.   (currently amended): A method for dynamically determining the health of a service resident on a host machine, comprising:

collecting service performance information from the resident service, wherein the collected service information relates to a plurality of performance metrics; and

translating the collected service performance information into a generic output relating to current operational performance of the service, wherein the generic output is one of a scriptable interface and an application programming interface, and useable by different performance monitoring tools,

wherein the generic output comprises a plurality of service health metrics, and wherein the translating step comprises combining one or more of the plurality of performance metrics to provide one or more of the plurality of service health metrics, and wherein the plurality of service health metrics comprises availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction.

2.   (previously presented):   The method of claim 1, wherein the host machine comprises one or more components, further comprising:

collecting external performance information from one or more of the one or more components;

translating the collected external performance information; and

combining the translated external performance information and the translated service performance information to provide the generic output.

3.   (cancelled):

4.   (original):   The method of claim 1, further comprising accessing the generic output to read the health of the service.

5.   (original):   The method of claim 1, wherein the collecting step comprises reading performance information provided by the service.

i

6.   (original):   The method of claim 1, wherein the collecting step comprises deriving performance information from the service.

7.   (original):   The method of claim 6, wherein the deriving step comprises using a wrapper program to read the performance information.

8.   (original):   The method of claim 6, wherein the deriving step comprises using a probe program to read the performance information.

9.   (cancelled):

10.   (cancelled):

11.   (currently amended): An apparatus that determines a health of a service resident on a host machine, comprising:

a data collection engine that collects service health information; and

a translation data analysis engine that translates the collected service health information using a health generation algorithm and provides one or more generic health metrics relating to current operational performance of the service, wherein the generic output is one of a scriptable interface and an application programming interface, and useable by different performance monitoring tools, wherein the collected service health information relates to a plurality of performance metrics, wherein the generic output comprises a plurality of service health metrics, and wherein the translation data analysis engine combines one or more of the plurality of performance metrics to provide one or more of the plurality of service health metrics, and wherein the plurality of service health metrics comprises availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction.

12.   (previously presented):   The apparatus of claim 11, wherein the host machine comprises one or more external components, wherein the data collection engine collects external performance information from one or more of the one or more external components information, and wherein the translation data analysis engine translates the collected external information using the health generation algorithm to provide the one or more generic health metrics.

ii

13.   (cancelled):

14.   (previously presented):   The apparatus of claim 11, wherein the data collection engine, comprises:

      a data query module that reads performance information from the service; and

      a data derivation module that derives performance information from the service.

15.   (original):   The apparatus of claim 14, wherein the data derivation module derives the performance information from one or more of a wrapper program, a benchmark program and a probe program.

16.   (original):   The apparatus of claim 11, wherein the health generation algorithm comprises:

      a weighting scheme that weights one or more performance information parameters;

      a summation scheme that combines one or more performance information parameters; and

      a averaging scheme that averages collected service health information for a service health metric.

17.   (original):   The apparatus of claim 11, further comprising an interval control engine that receives the service health information at a first time interval and provides an output having a second time interval different from the first time interval.

18.   (currently amended): A method for monitoring health data of a service operating on a host machine, comprising:

      collecting service performance information from the service;

      collecting external performance information from components of the host machine;

      translating the collected service and external performance information according to a health generation algorithm to generate a generic service health output; and

      providing the generic service health output relating to current operational performance of the service as an output file accessible by performance monitoring tools, wherein the generic service health output is one of a scriptable interface and an application programming interface, and usable by the different performance monitoring tools, wherein the collected service information relates to a plurality of performance metrics, wherein the generic output

comprises a plurality of service health metrics, and wherein the translating step comprises combining one or more of the plurality of performance metrics to provide one or more of the plurality of service health metrics, and wherein the plurality of service health metrics comprises availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction.

19.   (original):   The method of claim 18, wherein the step of collecting the service performance information comprises reading first service performance parameters and deriving second service performance parameters, and wherein the step of collecting the external performance information comprises reading first external performance parameters and deriving second external performance parameters.

20.   (previously presented):   The method of claim 18, further comprising collecting the service performance information on a first time interval and adjusting the first time interval to provide the generic service health output at a second time interval.

21.   (currently amended): An apparatus that determines a health of a service, wherein the service operates on a host computer, comprising:

        a collection module that receives performance information related to the service;

        a translation health generator module that applies a rule set to the received performance information and derives generic health metrics therefrom; and

        an output module that outputs the generic health metrics relating to current operational performance of the service, wherein the generic health metrics are in a format usable by different performance monitoring tools, wherein the generic health metrics comprise availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction.

22.   (original):   The apparatus of claim 21, wherein the collection module receives external performance information from one or more external services coupled to the host computer and receives internal performance information related to operation of the service on the host computer.

23.   (cancelled):

24.   (previously presented):   The apparatus of claim 21, wherein the generic health metrics is one of a scriptable interface and an application programming interface.

IX. EVIDENCE APPENDIX

No evidence submitted.

X.   RELATED PROCEEDINGS APPENDIX

No related proceedings.